User's Manual

NEC

QB-MINI2
Setup Manual

Partner Tool

Target Devices
V850 Microcontrollers

Document No. U19158EJ1V0UM00 (1st edition)
Date Published March 2008 NS

© NEC Electronics Corporation 2008
Printed in Japan

[MEMO]

MINICUBE is a registered trademark of NEC Electronics Corporation in Japan and Germany and Germany or a trademark in the United States of America.
Windows is either a registered trademarks or a trademark of Microsoft Corporation in the United States and/or other countries.
Green Hills Software and MULTI are a trademark of Green Hills Software, Inc. in the United States.
All other trademarks or registered trademarks are the property of their respective owners.

- **The information in this document is current as of March, 2008. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".

  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

  "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.

  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).

  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)

(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.

(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02. 11-1

INTRODUCTION

Readers

This manual is intended for engineers who use QB-MINI2 (alias: MINICUBE2) as well as MULTI, manufactured by Green Hills Software (hereinafter referred to as "MULTI") for debugging.
Engineers who read this manual are presumed to have a good knowledge of the functions of the device and how to use it as well as debuggers.

Purpose

The purpose of this manual is to help users understand the basic ways of using MINICUBE2 and MULTI when introducing them.

Organization

This manual is divided into the following sections.

- General
- Software tool installation
- User program preparation
- System start
- Multi debugger start
- Interrupt source names and start options
- MINICUBE2 self-diagnosis and firmware update

How to Read This Manual

It is assumed that the readers of this manual have general knowledge in the fields of electrical engineering, logic circuits, and microcontrollers.
This manual describes the basic setup procedures.

To understand the basic specifications and usages of MINICUBE2
→Read this manual according to the **CHAPTER 1 GENERAL**.

To know the manipulations, command functions, and other software-related settings of MINICUBE2
→See the user's manual of the debugger to be used.

Conventions

Note:	Footnote for item marked with **Note** in the text
Caution:	Information requiring particular attention
Remark:	Supplementary information
Numeric representation:	Binary ... xxxx or xxxxB
	Decimal ... xxxx
	Hexadecimal ... xxxxH
Prefix indicating power of 2 (address space, memory capacity):	K (kilo): 2^{10} = 1,024
	M (mega): 2^{20} = $1,024^2$

Terminology

The meanings of the terms used in this manual are described in the table below.

Term	Meaning
MINICUBE2	Generic name of QB-MINI2
Target device	This is the device to be emulated.
Target system	This is the system to be debugged (user-created system). Refers to all hardware and software provided by the user.
MULTI	Integrated development environment MULTI, manufactured by Green Hills Software

Related Documents

Please use the following documents in conjunction with this manual.
The related documents listed below may include preliminary versions. However, preliminary versions are not marked as such.

Documents Related to Development Tools (User's Manuals)

Document Name	Document Number
QB-MINI2 On-Chip Debug Emulator with Programming Function User's Manual	U18371E
QB-MINI2 Operating Precautions	ZUD-CD-07-1212-E
QB-Programmer Programming GUI Operation User's Manual	U18527E
MINICUBE2 Diagnostic Tool User's Manual	U18588E
MINICUBE OCD Checker User's Manual	U18591E

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document for designing, etc.

CONTENTS

CHAPTER 1 GENERAL 8

CHAPTER 2 SOFTWARE TOOL INSTALLATION 9

CHAPTER 3 USER PROGRAM PREPARATION 10

3. 1 Securing Memory Space and Setting Security ID 10
3. 2 Securing Serial Interface for Communication 15

CHAPTER 4 SYSTEM START 16

CHAPTER 5 MULTI DEBUGGER START 18

APPENDIX A INTERRUPT SOURCE NAMES AND START OPTIONS 23

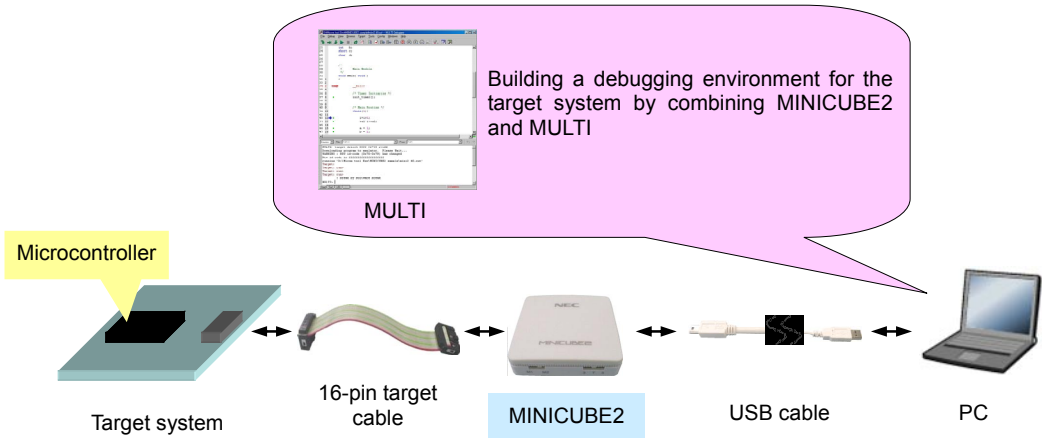
APPENDIX B MINICUBE2 SELF-DIAGNOSIS AND FIRMWARE UPDATE 24

CHAPTER 1 GENERAL

MULTI, manufactured by Green Hills Software, provides a high-performance and user-friendly integrated development environment, based on a unified GUI. Programs can be developed efficiently when MULTI is used together with on-chip debug emulator MINICUBE2 with a programming function.

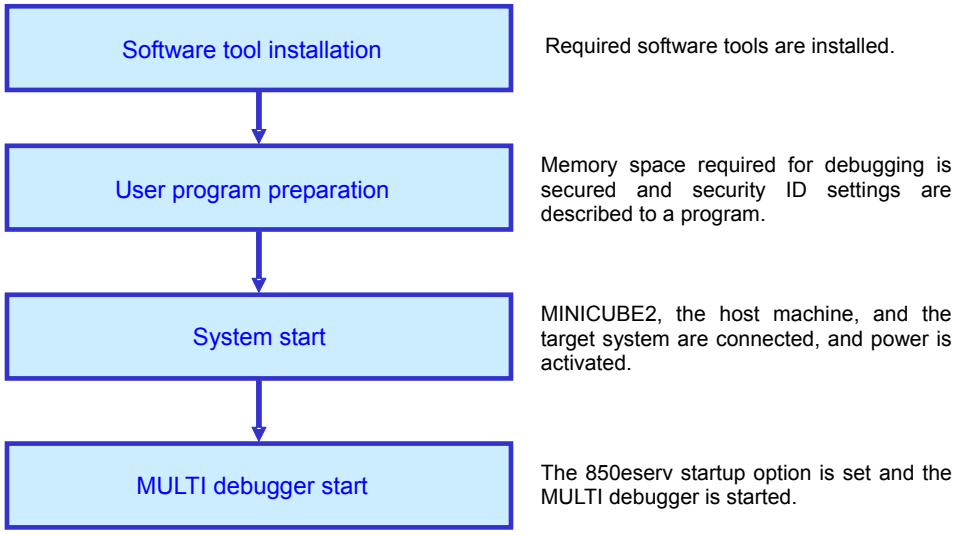
A combined environment of MULTI and MINICUBE2 can be easily built by setting up a system according to this manual.

Figure 1-1. Environment By Combining MINICUBE2 And MULTI



An overview of the setup flow is shown below. The setup is described in the following chapter.

Figure 1-2. setup flow


Software tool installation
Required software tools are installed.
User program preparation
Memory space required for debugging is secured and security ID settings are described to a program.
System start
MINICUBE2, the host machine, and the target system are connected, and power is activated.
MULTI debugger start
The 850eserv startup option is set and the MULTI debugger is started.

CHAPTER 2 SOFTWARE TOOL INSTALLATION

Install the following software tools.

(1) MULTI debugger

This is the integrated development environment MULTI debugger manufactured by Green Hills Software.

Execute setup.exe to install the debugger. For details, see the document related to MULTI.

(2) 850eserv

This is the debug server used to connect MINICUBE2 and the MULTI debugger. It is included with integrated development environment MULTI and should be installed together with MULTI. Use the latest version of 850eserv, due to the relation with the Exec library, described in (3), below.

(3) Exec library

This is the dynamic link library called by 850eserv to control MINICUBE2.

Due to the relation with 850eserv, described in (2), above, download the latest version of the Exec library from the following Web sites and copy the complete set of extracted files to the same folder as MULTI.exe.

Japanese Web site: http://www.necel.com/micro/ghs/jpn/exec/execindex.html
English Web site: http://www.necel.com/micro/ghs/eng/exec/index.html

(4) USB driver for MINICUBE2

This USB driver is required to connect the host machine and MINICUBE2.

Download it from the same Web site as the Exec library, described in (3), above, and extract it.

A plug-and-play dialog will be displayed when MINICUBE2 and the host machine are connected after the USB driver file has been extracted. Specify as the destination for storing the USB driver the folder that was extracted before (MQB2ALL).

(5) Device file

This file holds device-specific information and is used by 850eserv.

Download it from the following Web sites and extract it. The device file name after extracting (.800 file extension) and the path information of the extracted folder are used for specifying options when starting the MULTI debugger. For details, see **CHAPTER 5 MULTI DEBUGGER START**.

Japanese version: http://www.necel.com/micro/ods/jpn/ → Click “Version-up Service”.
English version: http://www.necel.com/micro/ods/eng/ → Click “Version-up Service”.

CHAPTER 3 USER PROGRAM PREPARATION

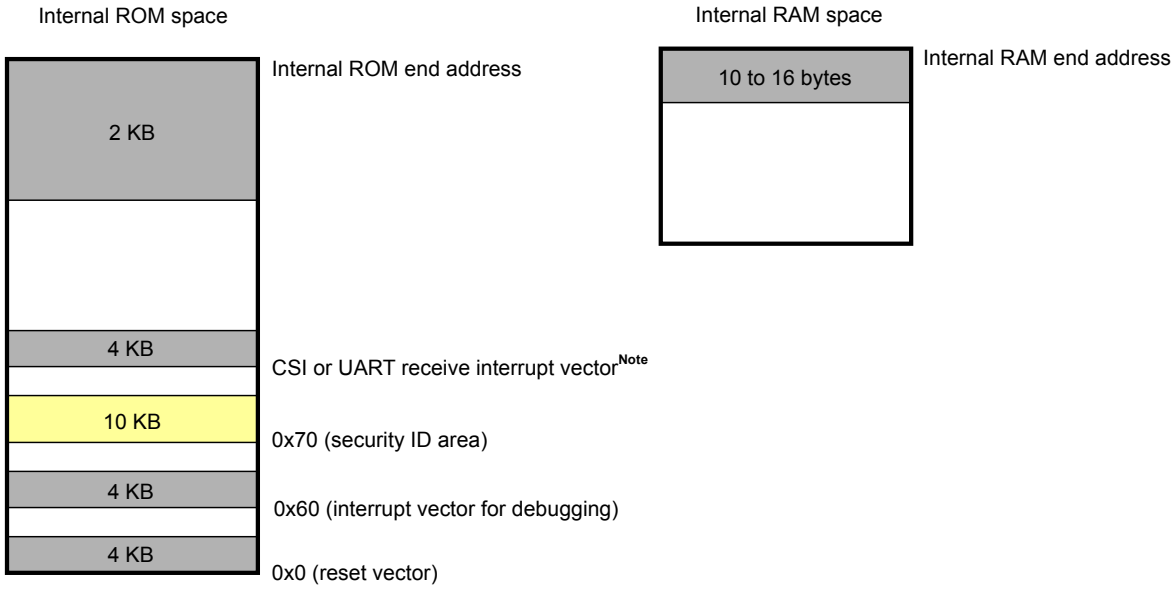
The following preparations must be performed for user programs so that MINICUBE2 can communicate with the target device and perform each debug function. These preparations require the user programs and linker directive to be edited. Refer to the following **3. 1 Securing Memory Space and Setting Security ID** to perform editing and building.

3. 1 Securing Memory Space and Setting Security ID

The shaded areas in **Figure 3-1** are areas to which user programs and data cannot be placed because a monitoring program for debugging will be built in. The 10-byte areas shown in yellow are areas to which ID codes are embedded so that the contents of a memory cannot be read by a third party. When using MULTI, MINICUBE2 can be started when the security ID and embedded ID match as the argument of 850eserv start option “-id”. Be sure, therefore, to surely manage the embedded security ID code. If the ID code is forgotten, the flash memory must be erased and the debugger must be restarted with the ID code set as “0xffffffffffffffffffff” (10 bytes).

To secure memory space and set the security ID, describe the assemble source and linker directive source, shown on the next page.

Figure 3-1. Memory Spaces Is Placed A Monitoring Program For Debugging



Note When there is a receive error interrupt or a receive status interrupt, the corresponding vector must be secured.

(1) Example 1 of securing memory space and setting the security ID

• Program description (Add the following as the assemble source.)

```
--### DBG0 Vector ###
.section ".DBG0", "ax"                  -- 0x60
.word 0xffffffff

--### Security ID ###
--### Set ID code arbitrarily. ###
.section ".S_CODE", "a"                 -- 0x70
.byte 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff

--### Serial Receive Interrupt Vector ###
.section ".INTCB0R", "ax"               -- Note
.word 0xffffffff

--### ROM area secured for MINICUBE2 MONITOR ###
.section ".MC2MON_ROM", "ax"
.rept 0x200
.word 0xffffffff
.endr

--### RAM area secured for MINICUBE2 MONITOR ###
.section ".MC2MON_RAM", "abw"
.global monitorramsym
monitorramsym:
.space 16
```

Note This description is applicable when CSIB0 is used as the interface of MINICUBE2 and the target device. Change the “INTCB0R” section according to the receive interrupt source of serial communication. See Appendix A for interrupt source names. They are also listed in the user’s manual of the target device. Refer to the following page when there is a receive error interrupt or a receive status interrupt.

- Linker directive description (Insert the following into the linker directive.)

  The following example is applicable when the internal ROM size is 256 KB and the internal RAM end address is 0x3ffefff.

```
MEMORY
{
        iROM    :   ORIGIN = 0x00000000 , LENGTH = 256k-2k
        MC2ROM  :   ORIGIN = 254k       , LENGTH = 2k
        iRAM    :   ORIGIN = 0x03FFC000, LENGTH = 12k-16
        MC2RAM  :   ORIGIN = 0x03FFEFF0, LENGTH = 16
}

SECTIONS
{
        .RESET          0x00000000      :> iROM
        .DBG0           0x00000060      :> iROM
        .S_CODE         0x00000070      :> iROM
        .INTCB0R        0x00000230      :> iROM    //Note
        .MC2MON_ROM                     :> MC2ROM
        .MC2MON_RAM                     :> MC2RAM
}
```

Note Use the source name described in the program as the interrupt source name (INTCB0R section). Also, describe as the address a vector address corresponding to the interrupt source. See the user's manual of the target device for vector addresses.

(2) Example 2 of securing memory space and setting the security ID (when there is a receive status interrupt during serial communication)

• Program description (Add the following as the assemble source.)

```
--### DBG0 Vector ###
.section ".DBG0", "ax"                  -- 0x60
.word 0xffffffff

--### Security ID ###
--### Set ID code arbitrarily. ###
.section ".S_CODE", "a"                 -- 0x70
.byte 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff

--### Serial Receive & Status Interrupt Vector ###
.section ".INTUD0S", "ax"               -- Note
.word 0xffffffff
.section ".INTUD0R", "ax"               -- Note
.word 0xffffffff

--### ROM area secured for MINICUBE2 MONITOR ###
.section ".MC2MON_ROM", "ax"
.rept 0x200
.word 0xffffffff
.endr

--### RAM area secured for MINICUBE2 MONITOR ###
.section ".MC2MON_RAM", "abw"
.global monitorramsym
monitorramsym:
.space 16
```

Note This description is applicable when UARTD0 is used as the interface of MINICUBE2 and the target device. Change the interrupt name according to the receive interrupt and receive status interrupt of serial communication. See Appendix A for interrupt source names. They are also listed in the user's manual of the target device.

- Linker directive description (Insert the following into the linker directive.)

  The following example is applicable when the internal ROM size is 256 KB and the internal RAM end address is 0x3ffefff.

```
MEMORY
{
        iROM        :       ORIGIN = 0x00000000 , LENGTH = 256k-2k
        MC2ROM      :       ORIGIN = 254k       , LENGTH = 2k
        iRAM        :       ORIGIN = 0x03FFC000, LENGTH = 12k-16
        MC2RAM      :       ORIGIN = 0x03FFEFF0, LENGTH = 16
}
SECTIONS
{
        .RESET              0x00000000          :> iROM
        .DBG0               0x00000060          :> iROM
        .S_CODE             0x00000070          :> iROM
        .INTUD0S            0x000002B0          :> iROM    //Note
        .INTUD0R            0x000002C0          :> iROM    //Note
        .MC2MON_ROM                             :> MC2ROM
        .MC2MON_RAM                             :> MC2RAM
}
```

Note Use the source names described in the program as the interrupt source names. Also, describe as the address a vector address corresponding to the interrupt source. See the user's manual of the target device for vector addresses.

3. 2 Securing Serial Interface for Communication

Create the user program, paying attention to the following items.

- Serial interface registers
  Do not use the user program for setting the registers related to UART and CSI-H/S, which are used for communication.
- Serial communication interrupt mask registers
  Generally, do not use the user program for changing the interrupt mask registers of UART and CSI-H/S, which are used for communication. To use the IMR register to change the interrupt mask settings in a batch, do not change the setting values of the target interrupt masks by a read-modify-write operation.
- Port-related registers
  To use UART for communication, do not perform register settings for ports such that the TxD and RxD pins become invalid. For CSI-H/S, do not perform register settings for ports such that the SI, SO, SCK, and H/S pins become invalid. The H/S pin is used as the port output for debugging.

Two examples are described below.

<Example 1> Settings other than the following are prohibited when the V850ES/KJ2 is the target device and UART0 is used.

	7	6	5	4	3	2	1	0
PFC3	x	x	x	x	x	x	0	0

x: Arbitrary

	7	6	5	4	3	2	1	0
PMC3L	x	x	x	x	x	x	1	1

x: Arbitrary

<Example 2> Settings other than the following are prohibited when the V850ES/HG2 is the target device and CSIB0 is used.

	7	6	5	4	3	2	1	0
PMC4	x	x	x	x	x	1	1	1

x: Arbitrary

	7	6	5	4	3	2	1	0
PMCCM	x	x	x	x	x	x	x	0

x: Arbitrary

	7	6	5	4	3	2	1	0
PMCM	x	x	x	x	x	x	x	0

x: Arbitrary

	7	6	5	4	3	2	1	0
PCM	x	x	x	x	x	x	x	Writing prohibited[Note]

x: Arbitrary

Note The monitoring program changes the port value corresponding to the H/S pin, according to the debugger status. When port register settings are to be manipulated in 8-bit units, there will be no problem if read modify writing is performed in the user program, but an unintended operation may result if an interrupt for debugging occurs before writing.

CHAPTER 4 SYSTEM START

Start the system by using the following procedure.

(1) MINICUBE2 switch setting

Set the mode selection switch to M2 (V850 microcontroller connection setting). Set the power supply selection switch, referring to the following table.

Caution Do not toggle the MINICUBE2 switch when a USB cable is connected. Remove the USB cable when toggling the switch.

Table 4-1. Power Supply Selection Switch Setting

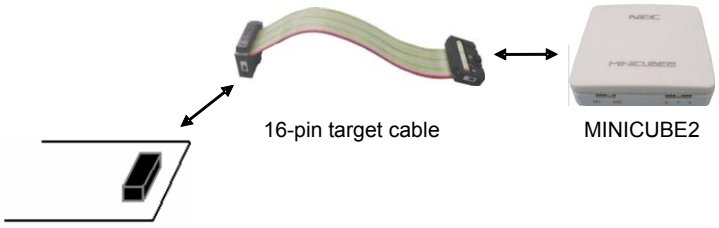
Setting	Description
3	Supplies 3 V from MINICUBE2 to the target system.[Note]
5	Supplies 5 V from MINICUBE 2 to the target system.[Note]
T	Uses the target system power supply. MINICUBE2 detects the target system power supply. Also, MINICUBE2 uses that power supply for the communication interface.

Note The maximum rated current is 100 mA. Do not use a target system exceeding this value. Also, power is always supplied when MINICUBE2 and the host machine are connected.

(2) Target system connection

Connect MINICUBE2 and the target system as shown in the **Figure 4-1**. At this time, the power of the target system must be turned off.

Figure 4-1. Connection of MINICUBE2 and Target System



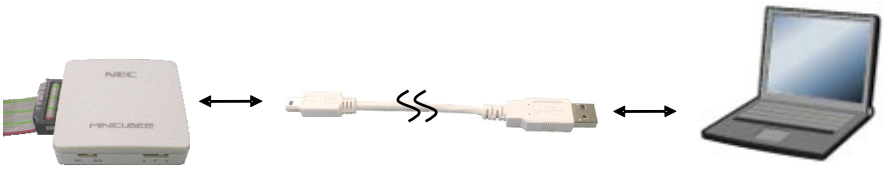
(3) USB connection

Connect MINICUBE2 and the target system as shown in the **Figure 4-2**. At this time, the power of the target system must be turned off.

When the power supply selection switch is set to “T”, the mode LED blinks in white after connection.

When the power supply selection switch is set to “3” or “5”, the mode LED glows steadily in white after connection.

Figure 4-2. Connection of MINICUBE2 and Host Machine



(4) Target system power activation

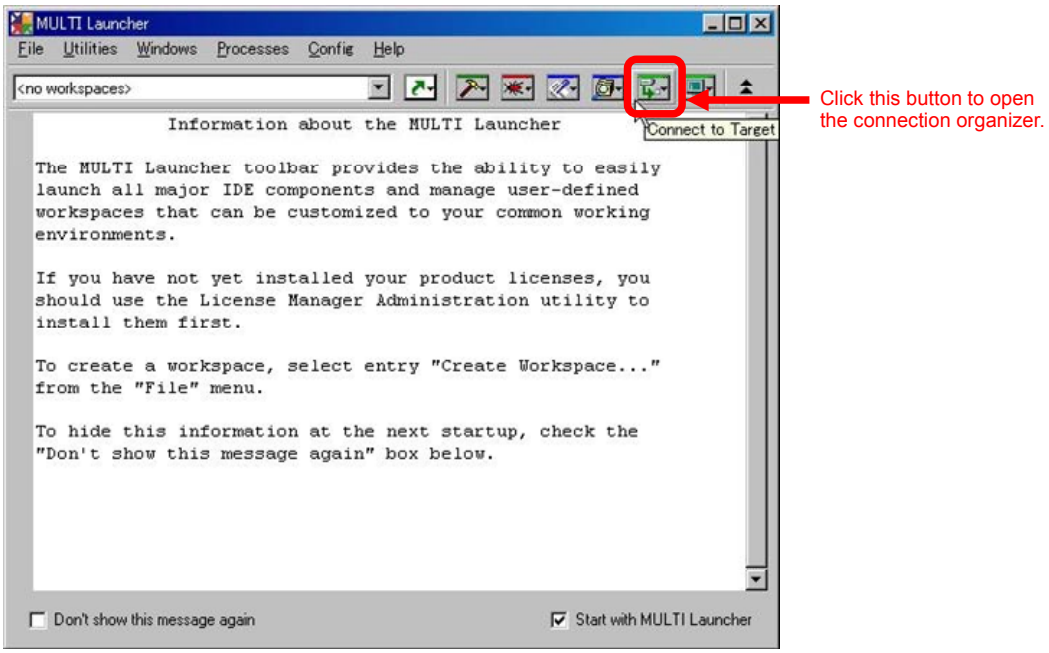
Activate the power of the target system. When the power supply selection switch is set to “3” or “5”, this step is not required. The mode LED glows steadily in white after power activation.

CHAPTER 5 MULTI DEBUGGER START

This chapter describes the procedure up to starting the MULTI debugger.

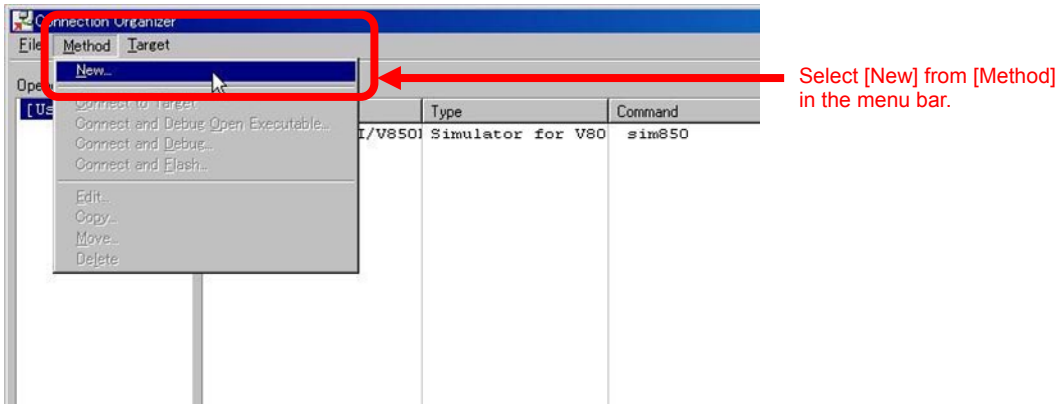
(1) Connection organizer start

Click the target connection button in the MULTI launcher to open the connection organizer, as shown below.


MULTI Launcher
File Utilities Windows Processes Config Help
<no workspaces>
Connect to Target
Information about the MULTI Launcher
The MULTI Launcher toolbar provides the ability to easily launch all major IDE components and manage user-defined workspaces that can be customized to your common working environments.
If you have not yet installed your product licenses, you should use the License Manager Administration utility to install them first.
To create a workspace, select entry "Create Workspace..." from the "File" menu.
To hide this information at the next startup, check the "Don't show this message again" box below.
Don't show this message again
Start with MULTI Launcher
Click this button to open the connection organizer.

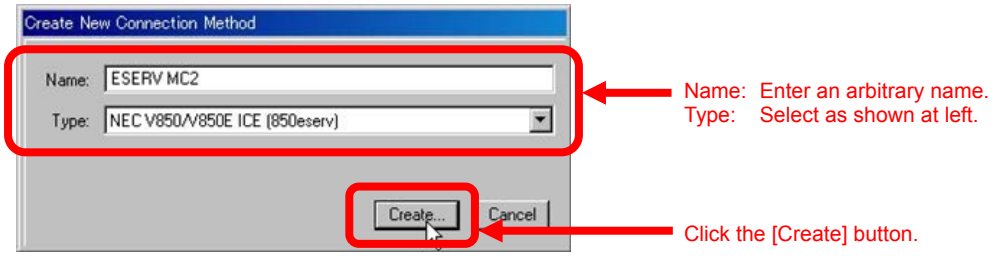
(2) Creating a new method

Select [New] from the [Method] menu in the connection organizer menu bar, as shown below.



(3) Setting a new method

Set the items in the new method dialog box and click the [Create] button, as shown below.



(4) Connection editor editing

Enter options in the [Other Options] field in the connection editor dialog box. See **Table 5-1** for option contents. The entry in the figure below is as follows (the tail of the entry is hidden in the figure).

-minicube2 –noint -p=csib0 -df=df3707.800 –id ffffffffffffffffffff -ip= D:¥device_file¥v850e

Click the [Connect] button after the options have been entered.

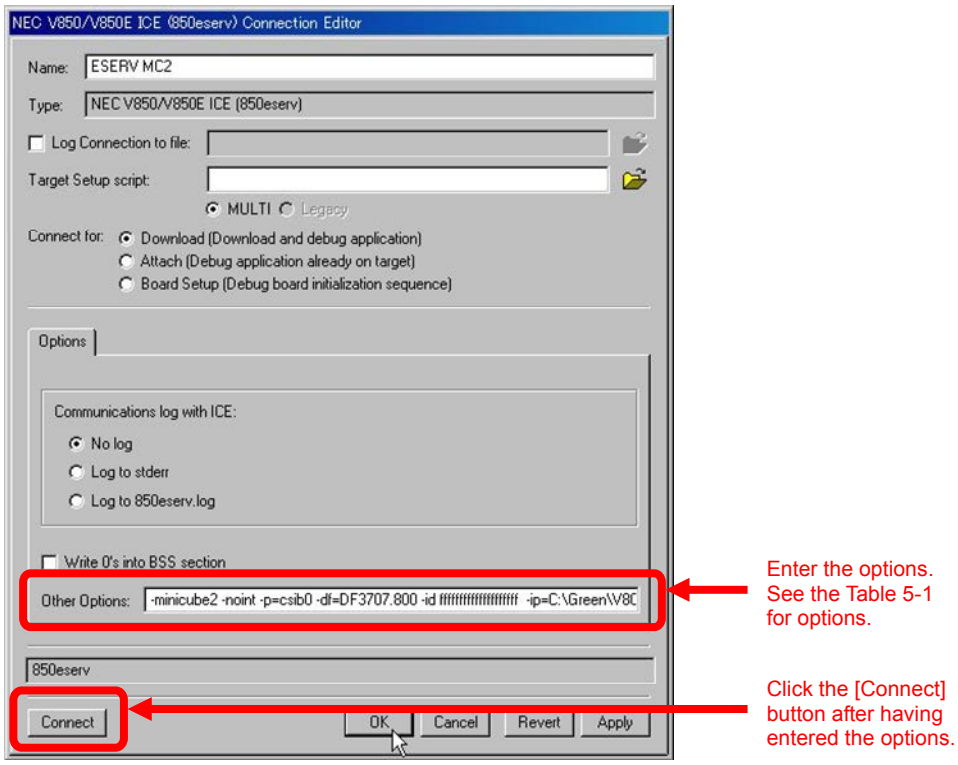


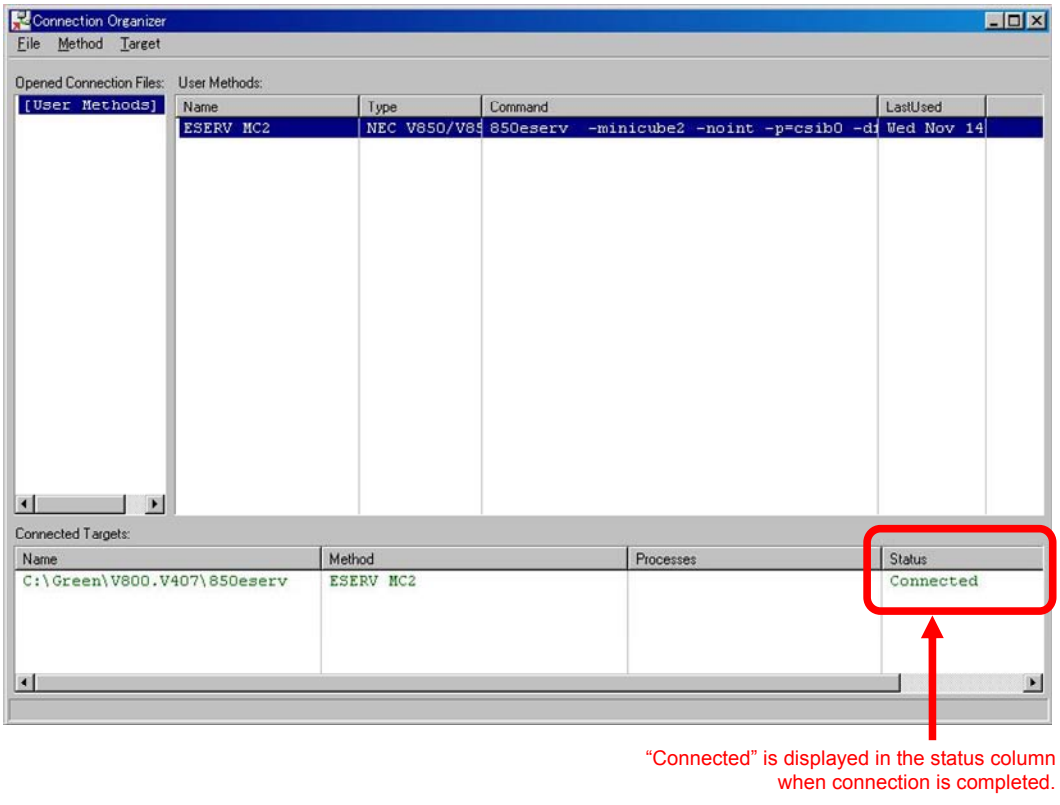
Table 5-1. Option Description

Option	Content	Necessity of Description
-minicube2	Option to be set when MINICUBE2 is the target.	Required option
-noint	Sets to non-interactive mode when a pio, a register, or an sfr command is issued. When non-interactive mode is set, a wait for input is not performed when the option is referenced.	Recommended option
-p	Specifies the name of the serial communication that connects MINICUBE2. Specify it by referring to **APPENDIX A INTERRUPT SOURCE NAMES AND START OPTIONS.**	Required option
-df	Specify the device file name (.800 file extension) according to the target device. See the device file user's manual (pdf) expanded in the folder generated when the device file was extracted (see **CHAPTER 2 (5) Device file**).	
-ip	Specify the folder in which the device file is stored.	
-id	Specify the security ID.	
-noiop	Option enabling referencing of programmable I/O area memories and changing memories. It is set when using the memory window to reference programmable I/O areas.	As required
-X0	Does not clear the BBS area to zero when downloading (default).	
-X1	Specify this option when clearing the BBS area to zero when downloading.	

(5) MINICUBE2 connection check

When the [Connect] button in the connection editor dialog box is clicked, 850eserve and MINICUBE2 communicate, and if a normal connection between them is confirmed, "Connected" will be displayed in the status column of the connection organizer, as shown below.

See the "V850E ICE Server Reference Manual" (document included in the MULTI environment set) when an error has occurred.



"Connected" is displayed in the status column when connection is completed.

The following error message will be displayed when the -p parameter is not set normally. In such a case, confirm whether the option descriptions of the connection editor are correct.

MULTI Debugger - MULTI v4.0.7 05_sep_26
Initializing `C:¥Green¥V800.V407¥850eserv -minicube2 -noint -df=DF3707.800 -id ffffffffffffffffffff -ip=D:¥device_file¥v850e'.
Establishing communication with remote.
Please wait...
"-minicube2" Error : please add "-p=<portname>".
Selectable port : UARTA0/CSIB0
Connection: No remote connection established.
OK

(6) MULTI debugger start and clock settings

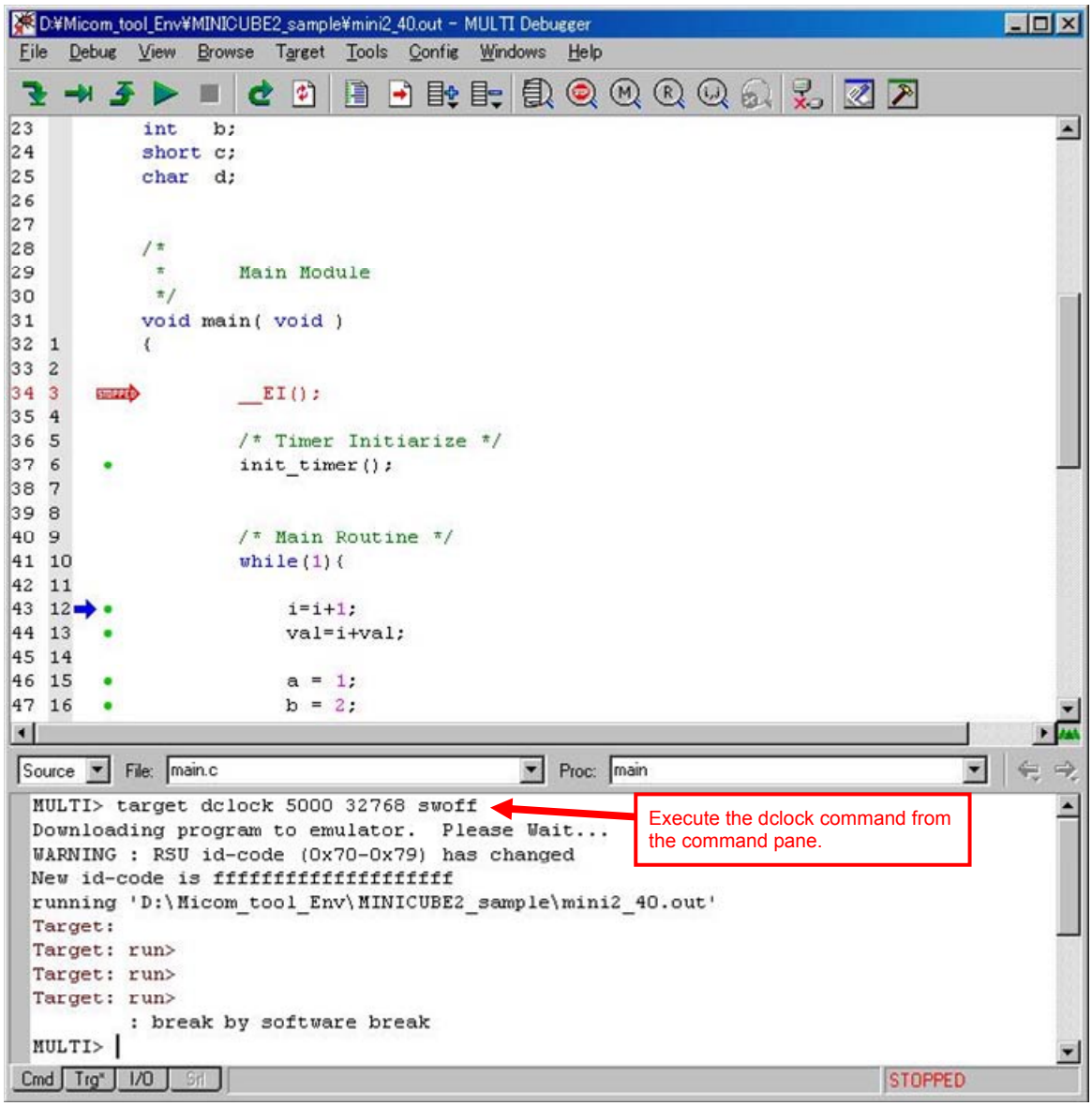
Start the MULTI debugger from the MULTI launcher.

Execute the dclock command from the command pane, as follows. In the example, the main clock is set to 5 MHz and the sub-clock is set to 32.768 kHz. Change the settings as required. Specify an oscillation clock (a clock before PLL multiplication) as the main clock.

MULTI > dclock 5000 32768 swoff

The base window of the debugger immediately after steps (1) to (6) have been executed and the program has been downloaded is shown below. See the following documents for the debug operations hereinafter.

- MULTI debugger command
  MULTI User's Guide Debugger
- 850eserv target command
  V850E ICE Server Reference Manual



APPENDIX A INTERRUPT SOURCE NAMES AND START OPTIONS

Target Device	MINICUBE2 Connection Serial Communication	Interrupt Vector	-p Option Specification
V850ES/Fx3	CSIB0	INTCB0R (CSIB0 receive complete interrupt)	csib0
	UARTD0	INTUD0S (UARTD0 status interrupt) INTUD0R (UARTD0 receive complete interrupt)	uartd0
V850ES/FG3L	CSIB0	INTUD0S (UARTD0 status interrupt) INTCB0R (CSIB0 receive complete interrupt)	csib0
	UARTD0	INTUD0R (UARTD0 receive complete interrupt)	uartd0
V850ES/Hx2	CSIB0	INTCB0R (CSIB0 receive complete interrupt)	csib0
	UARTA0	INTUA0R (UARTA0 receive complete interrupt)	uarta0
V850ES/Hx3	CSIB0	INTCB0R (CSIB0 receive complete interrupt)	csib0
	UARTD0	INTUD0R (UARTD0 receive complete interrupt)	uartd0
V850E/IA4	CSIB0	INTCB0RE (CSIB0 receive error interrupt) INTCB0R (CSIB0 receive complete interrupt)	csib0
	UARTA0	INTUA0RE (UARTA0 receive error interrupt) INTUA0R (UARTA0 receive complete interrupt)	uarta0
V850ES/IE2	CSIB0	INTCB0RE (CSIB0 receive error interrupt) INTCB0R (CSIB0 receive complete interrupt)	csib0
	UARTA0	INTUA0RE (UARTA0 receive error interrupt) INTUA0R (UARTA0 receive complete interrupt)	uarta0
V850ES/Ix3	CSIB0	INTCB0RE (CSIB0 receive error interrupt) INTCB0R (CSIB0 receive complete interrupt)	csib0
	UARTA0	INTUA0RE (UARTA0 receive error interrupt) INTUA0R (UARTA0 receive complete interrupt)	uarta0
V850ES/Jx2	CSIB0	INTCB0R (CSIB0 receive complete interrupt)	csib0
	CSIB3	INTCB3R (CSIB3 receive complete interrupt)	csib3
	UARTA0	INTUA0R (CSIB0 receive complete interrupt)	uarta0
V850ES/Jx3	CSIB0	INTCB0R (CSIB0 receive complete interrupt)	csib0
	CSIB3	INTCB3R (CSIB3 receive complete interrupt)	csib3
	UARTA0	INTUA0R (UARTA0 receive complete interrupt)	uarta0
V850ES/Kx1+	CSI0	INTCSI00 (CSIB0 receive complete interrupt)	csi0
	UART0	INTSR0 (UART0 receive complete interrupt)	uart0
V850ES/Kx2	CSI0	INTCSI00 (CSIB0 receive complete interrupt)	csi0
	UART0	INTSR0 (UART0 receive complete interrupt)	uart0
V850E/MA3	CSIB0	INTCSIR0 (CSIB0 receive complete interrupt)	csib0
	UARTA0	INTSR0 (UARTA0 receive complete interrupt)	uarta0

APPENDIX B MINICUBE2 SELF-DIAGNOSIS AND FIRMWARE UPDATE

When MINICUBE2 is not operating correctly, the MINICUBE utilities can be used to perform a self-diagnosis.
The MINICUBE utilities are also used when updating the internal firmware of MINICUBE2.
The MINICUBE utilities can be downloaded from the following Web sites.
See the included documents for how to use them.

Japanese Web site: http://www.necel.com/micro/ghs/jpn/exec/execindex.html
English Web site: http://www.necel.com/micro/ghs/eng/exec/index.html

[MEMO]

For further information, please contact:

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668,
Japan
Tel: 044-435-5111
http://www.necel.com/

[America]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
http://www.eu.necel.com/

Hanover Office
Podbielskistrasse 166 B
30177 Hannover
Tel: 0 511 33 40 2-0

Munich Office
Werner-Eckert-Strasse 9
81829 München
Tel: 0 89 92 10 03-0

Stuttgart Office
Industriestrasse 3
70565 Stuttgart
Tel: 0 711 99 01 0-0

United Kingdom Branch
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908-691-133

Succursale Française
9, rue Paul Dautier, B.P. 52
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

Sucursal en España
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

Tyskland Filial
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 638 72 00

Filiale Italiana
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02-667541

Branch The Netherlands
Steijgerweg 6
5616 HS Eindhoven
The Netherlands
Tel: 040 265 40 10

[Asia & Oceania]

NEC Electronics (China) Co., Ltd
7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian
District, Beijing 100083, P.R.China
Tel: 010-8235-1155
http://www.cn.necel.com/

Shanghai Branch
Room 2509-2510, Bank of China Tower,
200 Yincheng Road Central,
Pudong New Area, Shanghai, P.R.China P.C:200120
Tel:021-5888-5400
http://www.cn.necel.com/

Shenzhen Branch
Unit 01, 39/F, Excellence Times Square Building,
No. 4068 Yi Tian Road, Futian District, Shenzhen,
P.R.China P.C:518048
Tel:0755-8282-9800
http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.
Unit 1601-1613, 16/F., Tower 2, Grand Century Place,
193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: 2886-9318
http://www.hk.necel.com/

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-8175-9600
http://www.tw.necel.com/

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
http://www.sg.necel.com/

NEC Electronics Korea Ltd.
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737
http://www.kr.necel.com/

G0706